Amendments to the Claims:

A listing of the entire set of pending claims (including amendments to the claims, if any) is submitted herewith per 37 CFR 1.121. This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) A method of generating a linear transformation matrix A for use in a symmetric-key cipher, the method including:

[[-]] generating a binary [n,k,d] error-correcting code, represented by a generator matrix G $\in Z_2^{k \times n}$ in a standard form $G = (I_k \| B)$, with $B \in Z_2^{k \times (n-k)}$, where $k < n < 2k$, and d is the minimum distance of the binary error-correcting code;

[[-]] extending matrix B with 2k-n columns such that a resulting matrix C is non-singular, and

[[-]] deriving matrix A from matrix C.

2. (Currently amended) A method as claimed in claim 1, wherein ~~the step of~~ extending matrix B with 2k-n columns includes:

in an iterative manner:

[[-]] ~~(pseudo-)~~randomly generating 2k-n columns, each with k binary elements;

[[-]] forming a test matrix consisting of the n-k columns of B and the 2k-n generated columns; and

[[-]] checking whether the test matrix is non-singular,

until a non-singular test matrix has been found; and

using the found test matrix as matrix C.

3. (Currently amended) A method as claimed in claim 1, wherein ~~the step of~~ deriving matrix A from matrix C includes:

[[-]] determining two permutation matrices $P_1, P_2 \in Z_2^{k \times k}$ such that all codewords in an [2k,k,d] error-correcting code, represented by the generator matrix $(I \| P_1 C P_2)$, have a predetermined multi-bit weight; and

[[-]] using $P_1 C P_2$ as matrix A.

4. (Original) A method as claimed in claim 3, wherein the cipher includes a round function with an S-box layer with S-boxes operating on m-bit sub-blocks, and the minimum predetermined multi-bit weight over all non-zero codewords equals a predetermined m-bit weight.

5. (Currently amended) A method as claimed in claim 3, wherein ~~the step of~~ determining the two permutation matrices P_1 and P_2 includes iteratively generating the matrices in a ~~(pseudo-)~~ random manner.

6. (Original) A method as claimed in claim 1, wherein the cipher includes a round function operating on 32-bit blocks and wherein the step of generating a [n,k,d] error-correcting code includes:

generating a binary extended Bose-Chaudhuri-Hocquenghem (XBCH) [64, 36, 12] code; and

shortening this code to a [60, 32, 12] shortened XBCH code by deleting four rows.

7. (Original) A computer program product, wherein the program product is operative to cause a processor to perform the method of claim 1.

8. (Currently amended) A system for cryptographically converting an input data block into an output data block; the data blocks comprising *n* data bits; the system including:

[[-]] an input for receiving the input data block;

[[-]] a storage for storing a linear transformation matrix *A*, generated according to the method of claim 1,

[[-]] a cryptographic processor performing a linear transformation on the input data block or a derivative of the input data block using the linear transformation matrix *A*; and

[[-]] an output for outputting the processed input data block.

9. (New) A method comprising:

encrypting an input data block via a plurality of rounds to produce an output data block,

some or all rounds of the plurality of rounds including:

combining a first sub-block of the input data block with a key to provide a first output,

performing a non-linear permutation of the first output to produce a second output,

linearly transforming the second output to form a third output, and

providing the third output as a second sub-block to a next round,

wherein

linearly transforming the second output to form the third output includes an invertible transformation represented by a non-singular binary matrix that is based on a binary linear error-correcting code.

10. (New) A cryptographic engine, comprising:

a key addition module that is configured to combine input data with a key to form a first output,

a substitution module that is configured to perform a non-linear substitution of the first output to form a second output, and

a linear transformation module that is configured to perform a linear transformation of the second output to form a third output;

wherein

the linear transformation module is configured to perform the linear transformation via an invertible transformation using a non-singular binary matrix that is based on a binary linear error-correcting code.